TEST Engineering & Management

January - February 2020
ISSN: 0193 - 4120 Page No. 6189 - 6194

Feedback Design of a Mimo System using State Space Model

[1]A. Anilet Bala, [2]Sidhant Saxena, [3]Juili Gore, [4]Jayanta Ghosh
[1]Assistant Professor, [2,3,4]UG Scholars,
[1,2,3,4]Department of Electronics and Communication
[1,2,3,4]SRM Institute of Science and Technology, Chennai, Tamil Nadu, India

Article Info
Volume 82
Page Number: 6189 - 6194
Publication Issue:
January-February 2020

Article History
Article Received: *18 May 2019*
Revised: *14 July 2019*
Accepted: *22 December 2019*
Publication: *30 January 2020*

Abstract:
This project tends to stabilize the ambiguities that happen in several control systems which consists of the following process as follows: The microcontroller interfaces with the power board arm by sending pulse-width modulated (PWM) or analog signals to the driver and by reading sensors. Its output signal will be derived from sensor and motor readings in a feedback control loop. The control loop's behavior is adjusted by values/parameters sent to it from MPU6050 accelerometer using I^2C serial bus communication or python-based local server running on the computer. In addition, at regular intervals, the Arduino sends its measurements back to the Python server. The python-based local server running on your computer is an interface. It communicates with the Arduino via serial link, and with a Browser-based GUI. The Browser-based GUI plots measured data from the Arduino, passed to it by the local server, and provides simple interface for users to change the Arduino feedback loop parameters, which it passes on to the local server.

Keywords: *PWM, feedback control loop, MPU6050 accelerometer, I^2C serial bus, python-based server.*

I. INTRODUCTION

Thermostats, cruise control, camera auto-focusers, scooter stabilizers, aircraft autopilots, audio amplifiers, maglev; all examples that demonstrate the pervasive role of feedback control [3] in engineering design. But not all controllers use feedback. There are three main things that need to be focused on: Controller, Feedback and Feedforward.

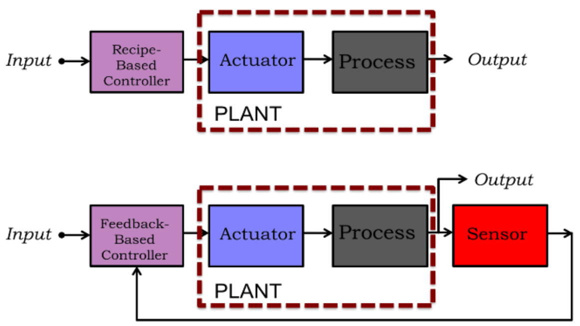


Fig.1 Feedback & Feed-forward systems

The *plant* [1], refers to the combination of what is being controlled, the *process*, and an instrument that modifies the process behavior, the *actuator* [1]. For example, the *process* could be room temperature, vehicle cruising speed, or camera focus, and the associated actuator could be the furnace in a temperature control system, the car engine in a cruise control system, or the lens-moving motor-and-gears in a camera autofocusing system. The *controller*, directs the actuator based on the input.

For example, if the input is a desired room temperature, then the controller determines when to start and stop the furnace so that the room stays at the desired temperature. If the controller bases its directions on measurements of the plant state, then we say it is a feedback controller, and if the directions are based on a predetermined recipe, we say it is a feed-forward controller.

The *sensor* block indicates an essential aspect of feedback control, that there is continuous sensing of the *state* of the plant. In a feedback controller, if the state is perturbed, the controller can easily adapt its directions to correct the state. For example, if one opens a window in a room with a feedback-based temperature controller,

January - February 2020
ISSN: 0193 - 4120 Page No. 6189 - 6194

the subsequent drop in temperature will be sensed by the controller, and the furnace will turn on to compensate.

In feed-forward control, there is no state monitoring. The controller sends instructions to the actuator by following a recipe. And that recipe is usually based on a model of how the plant will respond. Kicking a ball towards a goal is an example of feed-forward control, because your kick is based on a recipe, and that recipe is based on your mental model of how to get the ball to head towards the goal. Since you cannot redirect the ball in mid-flight, you are not using feedback control. In order to stabilize the whole system, we need to stabilize the natural frequencies that are being generated around the arms that will in turn tell us if the system is stable or not.

We chose homogenous LDEs so that we can emphasize three central concepts:

LDE's and the systems they represent have *natural frequencies*

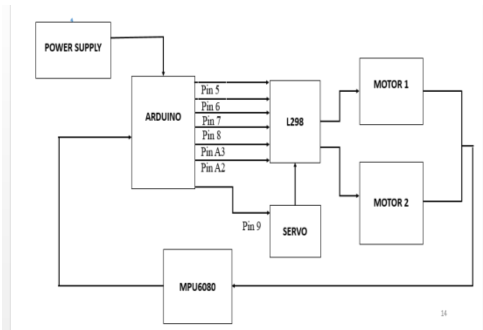
- If an LDE has a natural frequency with magnitude greater than one, the generated sequences will grow exponentially. We say that such LDE's are unstable.
- If a first-order LDE has a negative natural frequency, the generated sequences will oscillate. If the magnitude of a natural frequency is less than one, the LDE is stable and the oscillations decay to zero eventually. Otherwise, the LDE is unstable and the oscillations will grow.
- The natural frequency of a first-order LDE, or equivalently λ, governs the *evolution* of its associated sequences. In particular,
- If $\lambda>1$, the sequence grows monotonically and without bound, and we say that the first-order LDE is *unstable*.
- If $0<\lambda<1$, the sequence decreases monotonically to zero, and we say that the first-order LDE is *stable*.
- If $\lambda=1$, then the sequence does not change value with index, and the first-order LDE is neither *stable* nor *unstable*.

II. RELATED WORK

Masaki Inoue et al. [1] suggested the design of an optimal state feedback controller and real-time observer for twin rotor MIMO system (TRMS). Their objective is to construct a state feedback controller and observer for TRMS by determining the respective gains using the respective postulated algorithm, such as particle swarm optimization (PSO), which is capable of calculating desired trajectory along with better transient performances. Six states are considered in the state model of the TRMS model out of which only two are accessible, hence we require an efficient observer model. Performance is measured by the demonstrated results of the proposed controller and observer system. S.H. Zak et al. [2] and P. Biswas et al. [4] suggested particle swarm optimization (PSO) for the design of proportional-integral-derivative (PID) controller for a twin rotor MIMO (multi-input multi-output) system (TRMS). The objective of the paper is to adjust the gains of the PID controller automatically, through a universal search technique like PSO, so that the transient tracking error is minimal. The modeling of TRMS by Roshni Maiti et al. [3] exploits the nonlinear characteristics with the cross-coupling phenomena and efficient construction of PID control law for it, which is determined through the simulation environment. The proposed design approach is exercised to analyze various modes of operation of TRMS under different reference trajectories. Results show that the proposed design of PID controller is capable of determining different reference trajectories in subtle aspects.

III. APPROACH

Most modern controllers do not rely entirely on feedback, but use combinations of feed-forward and feedback control. The *plant* refers to the combination of what is being controlled, the *process*, and an instrument that modifies the process behavior, the *actuator*. The *sensor* block indicates an essential aspect of feedback control, that there is continuous sensing of the *state* of the plant. Feedback is responsible for magnitude and oscillation control & feedforward control is used for rapid and precise swing of copter arm.



TEST Engineering & Management

January - February 2020
ISSN: 0193 - 4120 Page No. 6189 - 6194

- PID Controller (here, Arduino UNO) accepts the control gains K_p, K_d and K_i.
- The output from the controller goes to the servo motor and the motor driver.
- Then, the motor driver (L298) controls the speed of the motor and simultaneously the angle is controlled through the servo motor.
- The motor speed and the angular position is fed back to the MPU6050 accelerometer,
- The accelerometer adopts the I2C protocol for high speed recovery mechanism. It determines the (x, y, z) co-ordinates and stabilizes the plant at the desired angular position.

IV. A STRATEGY FOR CONSTRUCTING A MATHEMATICAL MODEL FOR A SYSTEM OF INTEREST

PID (Proportional, Integral, and Derivative) controllers are ubiquitous in feedback systems, though for discrete-time systems, it would be more accurate to refer to such controllers as PSD (Proportional, Sum, Delta). To design PSD controllers, start with Proportional feedback with gain K_p. Then add Delta feedback, with gain K_d, to improve stability. And then, add Sum feedback, with gain K_i, to eliminate steady-state error.

K_P is the proportional constant that defines the magnitude at which the propeller rotates i.e. how much levitation the propeller attains and determines $\boldsymbol{\theta}$ (theta).

cos$\boldsymbol{\theta}$ is nonlinear function of $\boldsymbol{\theta}$, therefore proportional gain cannot be used for stabilizing the swing and rapid recovery of system. So, we introduce another controller gain constant K_i i.e. integral constant gain through feed forward technique.

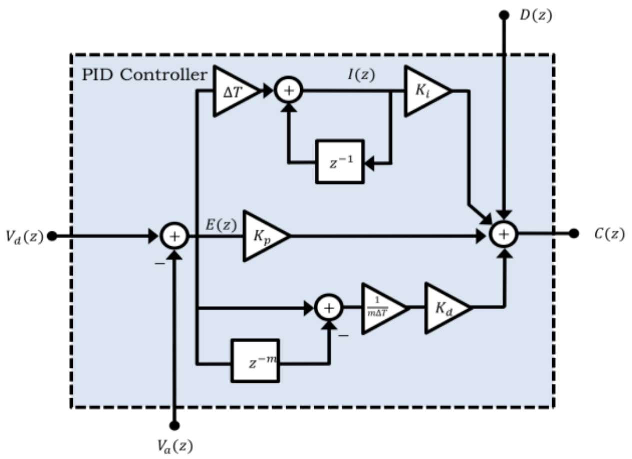


Fig.2 PID controller

Initially when the propeller starts, it rotates with a angular velocity ω(omega) due to which torque is produced that is proportional to the change in angular displacement. Perpendicular to this torque a propeller force f_m is generated which uplifts the arm. A similar and equivalent gravitational force f_g acts opposite to propeller f_m force.

$$F_{total} = F_{propeller} - F_g\cos\boldsymbol{\theta}$$

Here $F_{propeller}$ and cos$\boldsymbol{\theta}$ are variables. To stabilize the system we need to derive these functions. $F_{propeller}$ can oscillate on its position, hence another constant gain called $K_{d\,,}$ is given through feed back that controls these oscillations

V. SYSTEM MODEL

A. PLANT

The system is described by homogenous linear difference equations.

$$\omega[n] = \omega[n-1] + T\alpha[n-1]$$
$$\theta[n] = \theta[n-1] + T\omega[n-1]$$
$$\theta[n] - 2\theta[n-1] + \theta[n-2] = T^2\alpha[n-2]$$

Where, T is sampling period.
A desired angle $\theta_d[n]$ is fed to the PID controller and an error signal e[n] is generated by calculating the difference of desired and measured angle $\theta[n]$ from previous sample. This error signal is being set up by the PID control mechanism. The PID or PSD controller (due to discrete system) assembles a motor command m[n] through three different gain mechanisms:
K_p : proportional gain;
K_d :derivative gain;
K_i : integral gain.
Each entity has their particular significance.

B. CONTROLLER

1) Proportional control:

Kp is generally used to increase or decrease the motor current that goes into the propeller motor to drive the propeller. It directly controls the motor speed and angular velocity of propeller.

$$\boldsymbol{m[n] = K_p(\theta_d[n] - \theta[n]) + direct}$$

$$\boldsymbol{\theta[n] - 2\theta[n-1] + \left(1 + T^2\gamma K_p\right)\theta[n-2] = T^2\gamma K_P\theta_d[n-2] + T^2\gamma direct}$$

Due to air resistance and/or friction the motor current is always less than desired input. To increase the current

TEST Engineering & Management

January - February 2020
ISSN: 0193 - 4120 Page No. 6189 - 6194

weightage we need to increase K_p but a large increase in the proportional gain might lead to natural frequencies that are greater than unity. This leads the system to go unstable. Hence, proportional gain is itself not sufficient for stability system criterion (though feedforward approach can be implemented).

2) Integral control:

By introducing K_i , we accumulate all the previous error samples. On doing so, the motor current gets increased gradually and we need not increase K_p value. But, due to large accumulation the propeller rapidly jumps to a higher altitude and decelerate afterwards to the desired position. K_i is also added to remove steady state error. As n->∞$\alpha[n] = 0$.

3) Derivative control:

To compensate for the large accumulation of previous error samples, we apply derivative gain that neutralizes the effects of integral control and the system ends up in the desired position without any jitter.

$$m[n] = K_p e[n] + \frac{K_d}{T}(e[n] - e[n-1]) + K_i T \sum_{0}^{n} e[m]$$

$$e[n] = \theta_d[n] - \theta[n]$$

C. ACTUATOR

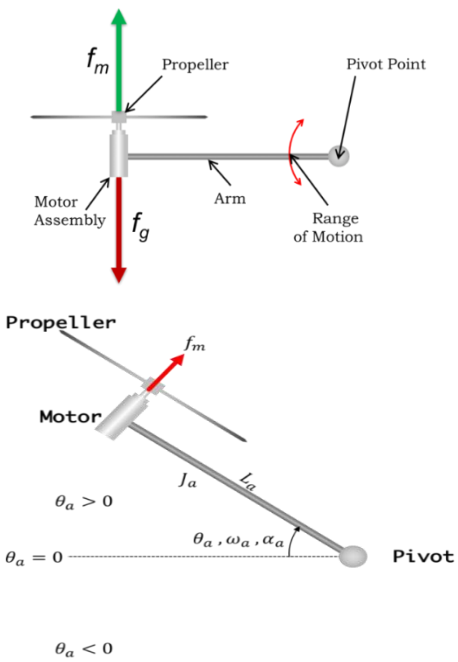


Fig.3 Balancing of propeller arm

To balance the propeller force we compute the model with gravitational force.

$$f = ma \qquad \alpha = \frac{a}{l}$$

Therefore; $\alpha = \frac{f}{ml}$

Where; m is mass of propeller assembly; l is length of arm.

To balance our arm;

$$f_{propeller} - f_g cos\theta = 0$$

Therefore; balancing factor

$$\gamma \approx -\frac{f_g \Delta\theta}{90}$$

This balancing factor is multiplied with the motor command to provide better input to the plant. i.e. $\alpha = \gamma m[n]$

VI. STATE SPACE MODEL

State space is a very common method of analyzing and controlling systems. One way of thinking of states is that they are the forms of energy that exist within a system. Another way of thinking about it is that it is the set of system values and conditions which unambiguously predict how a system will develop over time. In other words to have a model of a system predict its behavior, you generally need to specify initial conditions. These initial conditions are generally the states (or tightly related to the states through some constants).

As we go further and the number of states goes up they will get too bulky, and hard to analyze. A solution to this problem comes from realizing that all we're doing is linking a series of equations together. A group of equations can, when organized properly, be expressed as a matrix and with this in mind we can organize how we express our relationships in the system. We'll fit these relationships into two matrix-based equations of the following form which we'll call our state space equations, where A, B, C, and D are matrices and x is your state vector, u is your system input, and y is your system output. We'll use bold non-italic variables to represent matrices (to distinguish them from single variables which are italic and non-bold.:

x[n]=A·x[n−1]+B·u[n−1]
y[n]=C·x[n]+D·u[n]

The A matrix describes how the system states influence one another from time step to time step. In this matrix you will find relationships that come from physical laws and things like that (for example the relationship of

January - February 2020
ISSN: 0193 - 4120 Page No. 6189 - 6194

velocity to being the discrete time integral of acceleration may be found in here. A is basically the description of how the system would respond when left to its own devices. The matrix will have dimensions of m×m where m is the number of states.

The B matrix describes how the input influences the states of the system. Generally speaking the physical level of the input can sometimes "replace" a state from our system. By this I mean, if you have a system and your input signal is a force, as is in our rocket example, the state of acceleration (directly proportional to force applied) is now dictated by this input so we no longer treat it as a state. The B matrix will be m×1 in shape.

The C matrix describes how the output of the system (as defined by the user and or observer) is influenced by the states. If the output of interest is just one of the states (for example in our rocket problem), the C matrix will just contain a 1 in the location in the matrix corresponding to the state of interest. If the output depends on more than one state, then more than one value in the C matrix will be non-zero. Generally the C matrix will be 1×m in shape.

The D matrix describes how the input of the system u will directly influence the output of the system y. It will be a 1×1 matrix in our system.

The x vector contains all of the system's states in an order compatible with the matrices specified.

Often, the ability of an input u to immediately influence the output y will be negligible (usually there will be some delay and any change in input may have to propagate through the different states; in fact this disconnect between the input and output is part of the reason why we have to study controls in the first place). Therefore we'll generally ignore the D matrix unless we need it.

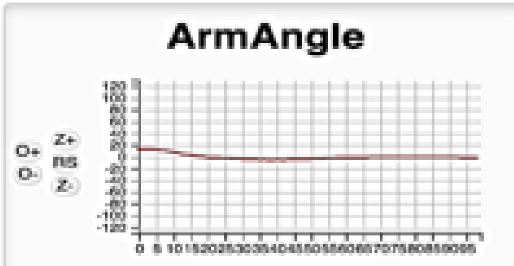
VII. RESULTS & DISCUSSIONS

This paper introduced a way to reduce uncertainties in the stability criterion of a closed loop system using state space model.

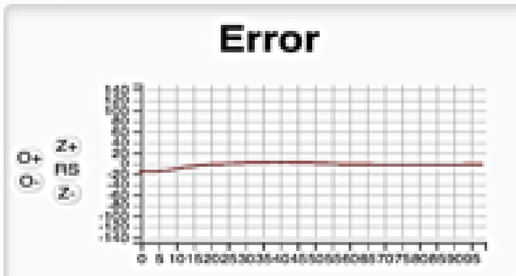
The twin rotor system efficiently works to provide unperturbed output and does not move its angular position unless the desired angle is changed. The system model is efficient in any desired environment.

The observer model produces real time results and parameters of the twin rotor system such as:

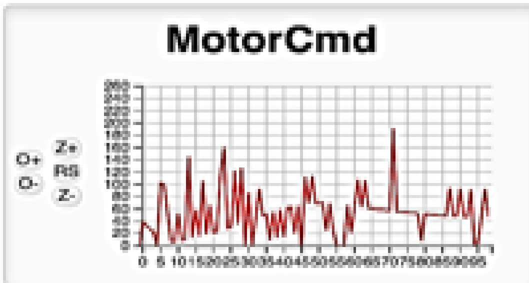
- Arm angle



- Error



- Motor Command



VIII. REFERENCES

1. State-space controller design for stabilization of descriptor systems by Masaki Inoue, Teruyo Wada, Masao Ikeda Department of Mechanical Engineering, Osaka University at SICE Annual Conference 2010 August 18-21, 2010, The Grand Hotel, Taipei, Taiwan.
2. S.H.Zak , B.L.Walcott & S.Hui, "Variable Structure Control &Observation of Non-Linear/Uncertain Systems" Variable Structure Control for Robotics & Aerospace applications young, K.K.K , Ed, Amsterdam, Elsevier Science Publishers, BV, pp.59-88(1993).
3. Roshni Maiti, Kaushik Das Sharma, Gautam Sarkar "Optimal State Feedback Controller & Observer Design for Twin Rotor MIMO System", 2nd International Conference on Control, Instrumentation,Energy & Communication (CIEC), 2016.

January - February 2020
ISSN: 0193 - 4120 Page No. 6189 - 6194

4. P. Biswas, R. Maiti, A. Kolay, K. Das Sharma, and G. Sarkar, “PSO Based PID Controller Design for Twin Rotor MIMO System”, in Proc. of International Conference on Control, Instrumentation, Energy and Communication (CIEC14), pp.106-110, January 2014, Kolkata, India. 2016 2nd International Conference on Control, Instrumentation, Energy & Communication (CIEC) 34.
5. H. V. Christensen, “Modelling and Control of a Twin-Rotor MIMO System,” Report of Department of Control Engineering Institute of Electronic Systems of Aalborg University, 2006.
6. Rahideh and M. H. Shaheed, “Mathematical Dynamic Modelling of a Twin-Rotor Multiple Input-Multiple Output System,” IMechE Journal of Systems and Control Engineering, vol. 221, no. 1, pp. 89–101, 2007.
7. J.J.E. Slotine, J.K. Hedrick, and E.A. Misawa “On Sliding Observers for Non-Linear Systems,” Trans. ASME J. Dyn. Syst. Meas. Contr., Vol. 109, No. 3, pp. 245-252 (1987).
8. C. Edwards, and S.K. Spurgeon, “On the Development of Discontinuous Observers,” Int. J. Contr., Vol. 59, No. 5, pp. 1211-1229 (1994).
9. Q. Ahmed, A.I. Bhatti, and S. Iqbal, “Nonlinear Robust Decoupling Control Design for Twin Rotor System,” 7th Asian Control Conference, Hong Kong, China, 2009.